REMARKS

Applicants express appreciation to the Examiner for the Interview of April 10, 2002, conducted with Applicants' attorneys. The Office Action of December 21, 2001 rejected claims 1-13, which remain pending. Applicants respectfully request favorable reconsideration of the pending claims in view of the amendments made herein and the matters discussed at the interview.

The Office Action rejected claims 1-4 and 10-13 under 35 U.S.C. §112, second paragraph as being indefinite. In response, claims 1 and 10 have been amended to correct the informalities identified in the Office Action.

The Office Action rejected claims 1-4 and 7-13 under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 5,872,956 to Beal and also by U.S. Patent No. 5,809,252 to Beighe. Claims 5 and 6 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Beal and also over Beighe.

At the Interview of April 10, 2002, Applicants' attorney discussed proposed amendments to claim 1. As discussed and agreed to at the interview as memorialized in the Interview Summary Record, the amendment made to claim 1 overcome the cited references.

Claim 1 as amended recites that:

the integrating component is positioned between the application and a connection-oriented device driver associated with the connection-oriented device.

The integrating component thereby enables the application to take advantage of the connection-oriented I/O subsystem and use the connection-oriented device using the known

application-level interfaces and without requiring the application programmer to program to an interface of the connection-oriented device driver.

In contrast, the component of Beal that the Office Action cited with respect to the integrating component is an "adapter component 108." The adapter component, however, is a sub-component of the device driver itself and interfaces with a network hardware adapter. (Fig. 1; col. 2, lines 7-9; col. 5, lines 12-14.) Moreover, Beal specifically states that the adapter component facilitates the process of developing device drivers themselves, which contrasts with the efficient application development that can be enabled by using the invention recited in claim 1. (Col. 5, lines 31-36.)

The Office Action points out that Beighe teaches separate paths for data packets and control packets. However, these separate paths are illustrated as being internal to a cable modem interface unit 90 (Fig. 3; col. 4, lines 42-47), and Beighe does not suggest or imply an integrating component or the interfaces that are recited in claim 1.

Thus, as discussed at the interview, claim 1 distinguishes from the cited references. The other independent claims 5, 7 and 10 include elements defining the position of the integrating component in a manner similar to that set forth above in reference to claim 1. Thus, these claims also distinguish from the cited references.

Applicants also point out that several of the claims have been amended to promote clarity, to provide terminology that is consistent with that of the base claims, and for other reasons that are not related to either responding to a rejection of the claims or distinguishing from cited art.

Attached hereto is a marked-up version of the changes made to the previous version of the specification and claims by this amendment. The attached pages are captioned "VERSION WITH MARKINGS TO SHOW CHANGES MADE."

For the foregoing reasons, Applicants submit that the pending claims are in condition for allowance and courteously request favorable action. If there are any outstanding issues that could be resolved by telephone, the Examiner is invited to contact the undersigned attorney.

Dated this 7 1 day of May, 2002.

Respectfully submitted,

R. BURNS ISRAELSEN
Attorney for Applicant
Registration No. 42,685

WORKMAN, NYDEGGER & SEELEY
1000 Eagle Gate Tower
60 East South Temple
Salt Lake City, Utah 84111
Telephone: (801) 533-9800
Facsimile: (801) 328-1707

022913

PATENT TRADEMARK OFFICE

RBI:llr
G:\DATA\WPDOCSRN\MICROSOF\OTHERDOC\73 amendment b.DOC

14

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION

The paragraph beginning at page 13, line 13 has been amended as follows:

To achieve the foregoing objects, and in accordance with the invention as embodied and broadly described herein a method, and computer program product for representing and connecting an underlying connection-oriented device in a known format is provided. [Note that co-pending] U.S. Patent Application Serial No. 09/097,293, filed [_____ (Filed] June 12, 1998, now issued as U.S. Patent No. 6,378,005, [) entitled "Method, Computer Program Product, and System for Separating Connection Management Functionality from a Connection-Oriented Device Driver,"] and U.S. Patent Application Serial No. 09/096,690, filed [_____ (Filed] June 12, 1998 [) entitled "Method and Computer Program Product for Managing Connection-Oriented Media"] are [herein] incorporated herein by reference.

IN THE CLAIMS

Claims 1-3, 5, 7, 8 and 10-12 have been amended as follows:

1.	(Twice Amended)	A method for representing to an application the characteristics of an underlying connection-oriented device over known application-level interfaces and allowing an application to take advantage of a connection-oriented I/O subsystem having an integrating component over the known application-level interfaces and without requiring the application programmer to program directly to the integrating component, the method comprising:

representing to an application, over a first known application-level interface associated with the integrating component, the connection control characteristics of the underlying connection-oriented device, wherein the integrating component is positioned between the application and a connection-oriented device driver associated with the connection-oriented device;

representing to the application, over a second known application-level interface associated with the integrating component, the data and data control characteristics of the underlying connection-oriented device;

15

receiving a [at least one] command from the application over [in] the first known application-level interface [format];

receiving a command from the application over the second known application-level interface; and

by the connection-oriented device driver, interacting with the integrating component [of the connection-oriented I/O subsystem] in order to [represent the underlying connection-oriented device characteristics over the known application-level interfaces and to] execute said received commands so that the [an] application may take advantage of the connection-oriented I/O subsystem and use the connection-oriented device using the known application-level interfaces and without requiring the application programmer to program to an [a new] interface of the connection-oriented device driver.

2.   (Amended)   A method as recited in claim 1 wherein the integrating component has a connection interface for making connections with underlying connection-oriented devices, and a data transport interface for interacting with a data transport component, the method further comprising [and the interacting with the integrating component comprises] the steps of:

the [having] data transport components interacting [interact] with applications and the data transport interface;

sending, to the integrating component, instructions [over the connection interface] for directing data and data control information over a specified data transport component; and

receiving, from the integrating component, an identifier that can be used by the application to access the data over the specified data transport component.

3.   (Amended)   A method as recited in claim 2 wherein:

the integrating component implements a connection manager interface that may support a connection manager component and the data transport components interact with the integrating component over the connection manager interface to effectively register their respective data types; and

the method further comprises the steps of:

[so that the] <u>receiving from the integrating component a</u> redirection command <u>specifying a data type</u> [received over the application-level interface specifying a data type]<u>; and</u>

[comprises the steps of] interacting over the connection manager interface of the integrating component in order to determine the correct data transport component based on data type.

5. (Amended) A connection-oriented driver subsystem where connection control information is communicated to an application through a connection [one] interface while data and data control information is communicated through a transport interface [driver], the driver subsystem comprising:

a [at least one simplified] connection-oriented device driver controlling a connection-oriented hardware [media] device;

a [at least one] data transport [protocol driver] capable of communication with an application;

an integrating component that interfaces with the [at least one simplified] connection-oriented device driver and the [at least one] data transport, said [at least one simplified] connection-oriented device driver and said [at least one] data transport serving as clients to said integrating [integration] component, wherein said integrating component is positioned between the application and the connection-oriented device driver, [and] said integrating [integration] component:

providing an abstracted connection interface that is available to a client that allows the [a] client to create a connection with a desired location using the [a] connection-oriented hardware [media] device [controlled by a simplified connection-oriented device driver]; and

providing facility for associating the [a] connection [created by a client through the connection interface] with the [a] data transport, thereby allowing the client to send and receive data and data control information over the [previously established] connection; and

a proxy client component [driver] that interfaces with the connection interface and the transport interface of the integrating [integration] component as a client, said proxy client component [driver]:

receiving abstract connection creation and control commands from the [an] application and implementing said [such] commands through [appropriate] use of the connection interface to create and manage the [a] connection;

causing redirection of data and data control information from the [previously created] connection through the proxy client component

18

[driver] to a designated <u>data</u> transport designated in <u>one of the</u> [an] abstract connection control <u>commands</u> [command]; and

returning to <u>the</u> [an] application, in response to a previously received connection control command, an identifier to be used by the application for receiving data and data control information from the designated <u>data</u> transport so that <u>the</u> connection control information is communicated to the application through the proxy <u>client component</u> [driver] while <u>the</u> data and data control information is communicated to the application through the designated <u>data</u> transport.

7.   (Twice Amended)   A computer program product for interacting with known application-level interfaces and an integrating component of a connection-oriented I/O subsystem in order to represent the characteristics of an underlying connection-oriented device to an application and allow an application to take advantage of the connection-oriented I/O subsystem over the known application-level interfaces without requiring the application programmer to program to a new interface, said computer program product comprising:

a computer-readable medium; and

computer-executable instructions carried on said computer-readable medium for performing the steps of:

representing the connection control characteristics of the underlying connection-oriented devices to an application over a first known application level interface associated with the integrating component, wherein the integrating component is positioned between the application and a connection-oriented device driver associated with the connection-oriented device;

representing the data and data control characteristics of the underlying connection-oriented devices to the application over a second known application level interface associated with the integrating component;

receiving a command from the [abstract connection creation and control commands from an] application over the first known application-level interface [interfaces];

receiving a command from the application in the second known application-level interface; and

by the connection-oriented device driver, interacting with the integrating component to execute said received [ascertain the underlying connection-oriented device and implement received connection creation and control] commands.

8.   (Amended)   A computer program product as recited in claim 7, wherein the command received from the application over the first known application-level interface comprises [further comprising computer-executable instructions for performing the steps of: receiving] a connection creation command, [in the known application-level interface format;] and wherein the computer-executable instructions operate to perform the steps of:

interacting with the integrating component to create the connection;

receiving a redirection command to send data and data control information received over the connection to a designated data transport;

causing redirection of data and data control information from the previously created connection to a designated transport; and

returning to the application an identifier to be used by the application for receiving data and data control information from the designated transport.

10.   (Amended)   A method for representing to an application the characteristics of an underlying connection-oriented device over known application-level interfaces and allowing an application to take advantage of a connection-oriented I/O subsystem having an integrating component over the known application-level interfaces and without requiring the application programmer to program directly to the integrating component, the method comprising:

separating connection control characteristics from data and data control characteristics received from an underlying connection-oriented device;

representing to an application, over a first known application-level interface associated with the integrating component, the connection control characteristics of the underlying connection-oriented device, wherein the integrating component is positioned between the application and a connection-oriented device driver associated with the connection-oriented device;

representing to the [an] application, over a second known application-level interface associated with the integrating component, the data and data control characteristics of the underlying connection-oriented device;

receiving a [at least one] command from the application over [in] the first known application-level interface [format];

receiving a command from the application over the second known application-level interface; and

by the connection-oriented device driver, interacting with the integrating component [of the connection-oriented I/O subsystem] in order to [represent the underlying connection-oriented device characteristics over the known application-level interfaces and to] execute said received commands so that the [an] application may take advantage of the connection-oriented I/O subsystem and use the connection-oriented device using the known application-level interfaces and without requiring the application programmer to program to an [a new] interface of the connection-oriented device driver.

11.   (Amended)   A method as recited in claim 10 wherein the integrating component has a connection interface for making connections with underlying connection-oriented devices, and a data transport interface for interacting with a data transport component, the method further comprising [and the interacting with the integrating component comprises] the steps of:

22

the [having] data transport components interacting [interact] with applications and the data transport interface;

sending, to the integrating component, instructions [over the connection interface] for directing data and data control information over a specified data transport component; and

receiving, from the integrating component, an identifier that can be used by the application to access the data over the specified data transport component.

12.   (Amended)   A method as recited in claim 11 wherein:

the integrating component implements a connection manager interface that may support a connection manager component and the data transport components interact with the integrating component over the connection manager interface to effectively register their respective data types; and

the method further comprises the steps of:

[so that the] receiving from the integrating component a redirection command specifying a data type [received over the application-level interface specifying a data type]; and

[comprises the steps of] interacting over the connection manager interface of the integrating component in order to determine the correct data transport component based on data type.